

8. Miguel

Program Name: Miguel.java

Input File: miguel.dat

Miguel has dreams of creating a very cool and very challenging game involving letters and a puzzle situation. He wants to encrypt a phrase and then have the user try to determine what the original was. Below is an example:

The question might be: What did the Oz girl say to her canine?

The clue would be: Aaaa, E'ee e eefgiii kl'mn nnn no Ooorrs stttvwy.

The answer is: Toto, I've a feeling we're not in Kansas anymore.

To start this game, Miguel needs a program that will take a quote, or any list of characters, and encrypt it using the following rules:

- If a character is any non-alphabetic character, it will remain in the same position.
- All of the letters will be sorted in alphabetical order not considering the case of the letter.
- Then, the sorted letters will be placed back in the list of characters in positions where letters originally were. If a position originally held an uppercase letter, the new letter in that position will now be uppercase. If a position originally held a lowercase letter, it will again be lowercase.

Look carefully at the example above as you plan your approach.

Miguel is counting on you to write this program so he can start making his millions of dollars as soon as possible.

Input: Input will consist of an integer N, the number of test cases. The number of test cases will be in range [1,20]. Each subsequent line will consist of a list of characters that includes both letters and non-letters. Any printed keyboard character could be used. The length of the string of characters is in the range [1,80]

Output: Each line of output will consist of a list of characters. All non-letters will be in their original positions. Every position that was an uppercase letter will still hold an uppercase letter, and every position that was a lowercase letter will still hold a lowercase letter. Now, however, the original letters have been sorted.

Sample input:

```
5
Dog
University of Texas Interscholastic League
Lmnop1234-5678Qrstabcdehijkl
z-C-N-e=I^q
Rock-Paper-Scissors
```

Sample output:

```
Dgo
Aaacceeeee fg Hiiii Llnnoorrsssstt Tuuvxy
Abcde1234-5678Fghijklmnopqrst
c-E-I-n=Q^z
Acce-Ikoop-Prrrssss
```